Predicting New Airbnb Users First Destination Country Using Machine Learning Model

EXECUTIVE SUMMARY

Airbnb's project aims to predict the country of a new user's first booking, providing personalized destination recommendations that enhance user engagement and expedite the booking process. Using demographic and activity data, this model leverages machine learning techniques, including Gradient Boosting, and XGBoost classifiers, to predict likely booking destinations. The project includes data preparation, feature engineering, model training, and evaluation, with NDCG@5 as the key metric for ranking accuracy. Expected outcomes are faster user engagement, improved demand forecasting, and increased conversion rates through personalized recommendations. Future projects could enhance the model with seasonal trends, segment users by travel type, and develop targeted marketing. Predictive insights from this model can also help Airbnb optimize regional resources and long-term demand forecasts, supporting personalized, data-driven experiences across expanding markets.

INTRODUCTION:

In the competitive and personalized landscape of travel and hospitality, predicting the first destination country of a new Airbnb user is essential for a tailored and engaging user experience. The project's goal is to leverage user data, such as demographics, device usage, browsing sessions, and engagement sources, to forecast the country of the first booking accurately. By doing so, Airbnb aims to optimize recommendations for new users, aligning suggested destinations with individual preferences and behaviors from the outset. Accurately identifying likely booking countries will not only enhance user satisfaction but also streamline the platform's ability to drive early engagement and increase the likelihood of bookings.

The scope of the project is to build a machine learning model that predicts a new user's initial booking destination country. The model will employ ensemble machine learning methods, namely Gradient Boosting, and XGBoost classifiers. These algorithms are selected for their effectiveness in handling large datasets with complex feature interactions, which are expected in user behavioral and demographic data. The project also includes detailed feature engineering to extract meaningful patterns from diverse data attributes like age, account signup details, language preferences, and session logs. Once the features are engineered, the models will be trained, tuned, and evaluated against multiple metrics, including precision, accuracy, and NDCG. Ultimately, this model will be used to create a ranked list of the top five likely countries, enhancing the recommendation system's ability to engage users early in their journey.

Personalization in digital platforms has become a cornerstone for user satisfaction, particularly in travel where options are vast and user preferences highly individualistic. For Airbnb, correctly predicting where a user will first book enables more relevant and appealing recommendations. By aligning content with likely destinations, Airbnb can increase the chances of users finding accommodations that fit their preferences, thereby reducing the time to the first booking and establishing a stronger relationship with the platform. Additionally, accurate predictions support Airbnb's operations and marketing strategies, as the company can better allocate resources and tailor regional campaigns based on anticipated demand. The motivation behind this project is to personalize user experiences effectively and to provide Airbnb with data-driven insights into user trends and behavior patterns.

Recommendation systems in travel have been significantly advanced by machine learning models, especially those based on ensemble methods. Gradient Boosting, and XGBoost are widely recognized for their capabilities in capturing complex relationships within high-dimensional data, making them ideal for scenarios where user behavior is influenced by multiple factors. In past projects involving user recommendations, ensemble techniques have shown considerable improvement in accuracy over individual models, thanks to their ability to reduce variance and minimize overfitting. Additionally, the application of XGBoost in real-world recommendation systems, such as in e-commerce and streaming services, has demonstrated its efficiency and scalability. By leveraging such techniques, this project builds on proven methodologies to maximize predictive performance and cater to Airbnb's need for quick and accurate user profiling.

Value Added

- **Enhanced Personalization**: By tailoring recommendations based on probable destinations, new users are more likely to find relevant listings, improving their satisfaction and engagement.
- **Improved Forecasting and Resource Allocation**: Accurate predictions about regional demand allow Airbnb to make informed decisions about resource distribution, promotions, and customer service in different geographic areas.'
- **Optimized Conversion Rates**: With faster and more precise recommendations, users may be more inclined to complete bookings, potentially increasing conversion rates and revenue.
- **Robust and Generalizable Predictive System**: The use of ensemble models ensures that the system can handle diverse user data characteristics and adapt to various user types, making the recommendation engine more resilient and applicable to a wide user base.
- **Competitive Advantage:** As more platforms adopt personalized recommendation systems, Airbnb's predictive model can provide a unique advantage by delivering a seamless and efficient onboarding experience that distinguishes it from competitors.

High-Level Approach

1. Data Collection and Preprocessing: Gather comprehensive user data, including demographic, session, and engagement data. The initial step also includes cleaning and standardizing data, handling missing values, and encoding categorical variables to prepare it for machine learning.

2. Feature Engineering: Develop and select features that could influence a user's booking choice. These might include specific device types, engagement sources (e.g., marketing channels), session duration, and other indicators of user intent. Feature engineering also involves testing combinations of features to enhance the model's ability to capture user patterns.

3. Model Selection and Training: Choose Gradient Boosting, and XGBoost as the main algorithms due to their robustness and adaptability to complex datasets. Train each model individually, using hyperparameter tuning to optimize them for predictive accuracy and efficiency, ensuring that the models generalize well on unseen data.

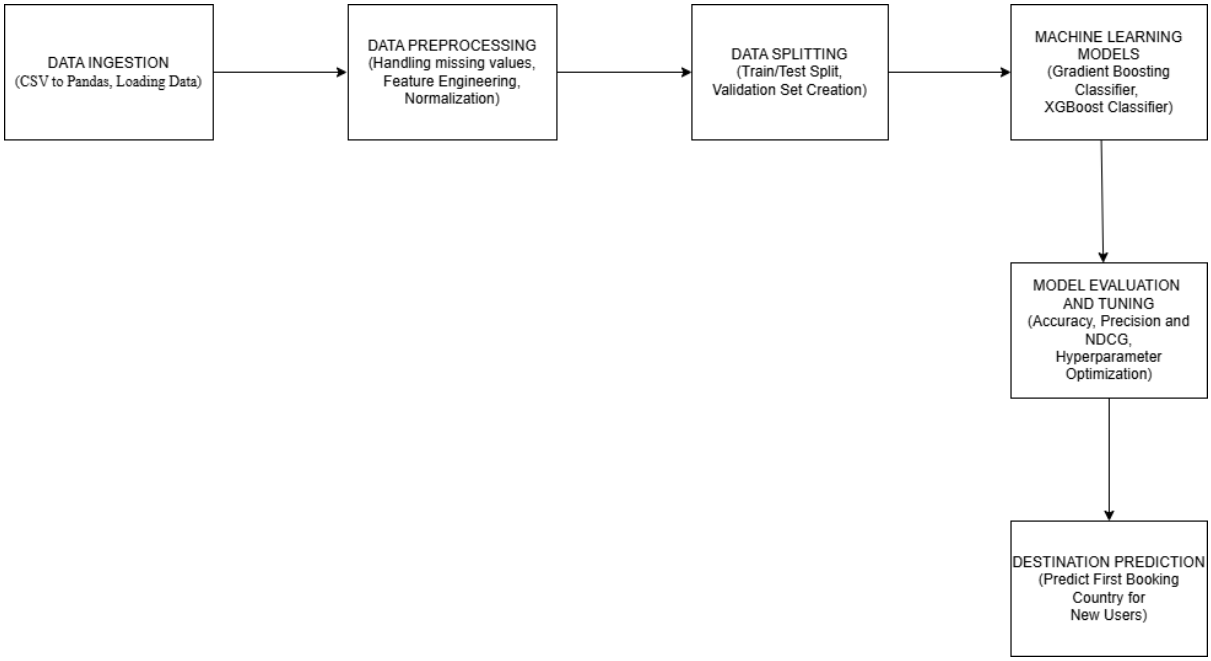
4. Evaluation and Comparison: Evaluate the models using metrics such as accuracy and precision, as these metrics provide a comprehensive view of the model's performance. The main criterion, however, will be the NDCG@5 score, as it reflects the model's ability to rank the most likely booking destinations within the top 5 options for each user.

5. Deployment and Integration: Once validated, the model will be integrated into Airbnb's recommendation system, allowing the platform to present users with relevant country recommendations

as soon as they join. This stage will include monitoring model performance and retraining periodically with new data to ensure recommendations remain accurate over time.

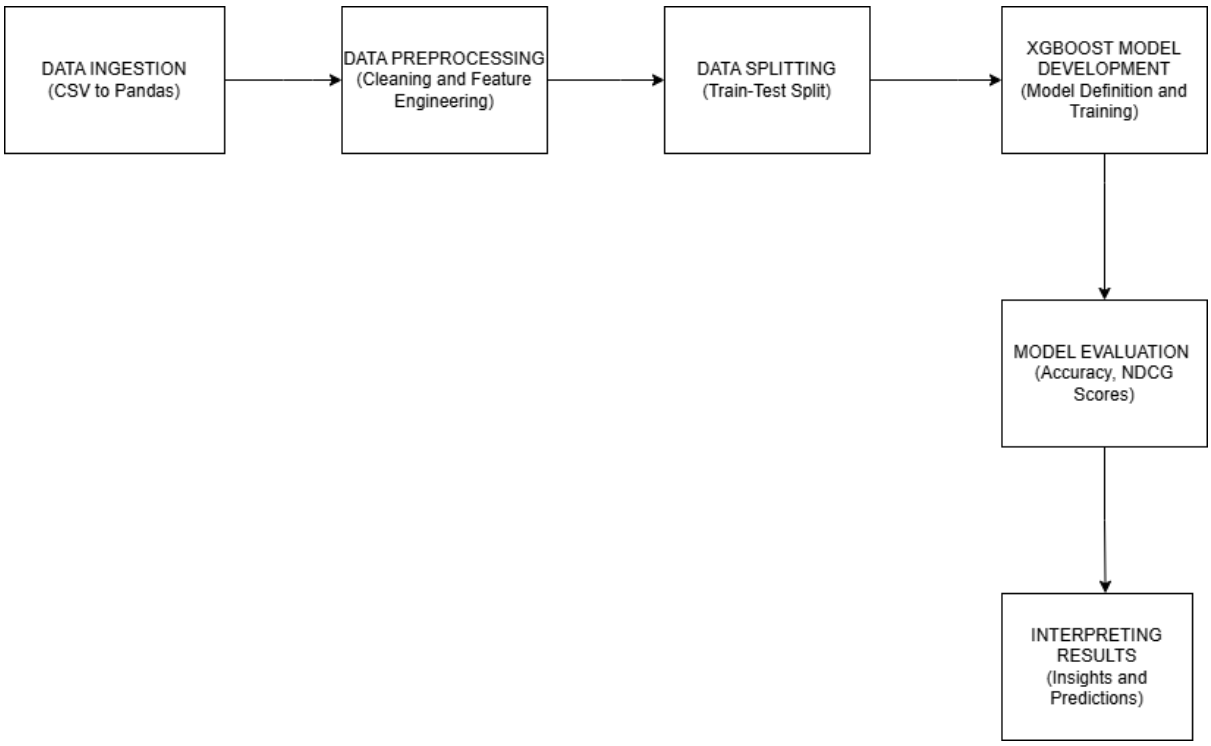
ARCHITECTURE:

Gradient Boosting Classifier:



The architecture diagram outlines a pipeline for predicting the first booking country for new users using machine learning models. It starts with Data Ingestion, where raw data is loaded from CSV files into a data processing framework like Pandas. Next, Data Preprocessing is performed, involving handling missing values, feature engineering, and normalization to prepare the data for modeling. The data is then split into training, validation, and test sets in the Data Splitting step to enable effective model training and evaluation. In the Machine Learning Models phase, algorithms such as Gradient Boosting Classifier and XGBoost Classifier are applied. The Model Evaluation and Tuning step follows, where models are assessed using metrics like accuracy, Precision, NDCG and undergo hyperparameter optimization for enhanced performance. Finally, the trained models are utilized in the Destination Prediction step to predict the first booking country for new users, providing valuable insights for user engagement and strategic decision-making.

XGBoost Classifier:



The architecture diagram outlines a pipeline for developing and evaluating an XGBoost model for predictive analysis. It begins with Data Ingestion, where raw data is loaded from CSV files into a data manipulation framework like Pandas. Next, Data Preprocessing involves cleaning the data to handle inconsistencies and conducting feature engineering to create useful features. In the Data Splitting step, the dataset is divided into training and test sets for effective model training and validation. The XGBoost Model Development phase includes defining and training the model, known for its efficiency and strong performance with structured data. Model Evaluation follows, where metrics such as accuracy and NDCG scores are used to assess the model's performance. Finally, in Interpreting Results, the model's predictions are analyzed to extract insights and inform decisions. This structured pipeline ensures thorough data preparation, robust model training, effective evaluation, and insightful result interpretation for practical applications.

MODELLING:

1. Data Properties

The dataset used in the Airbnb New User Bookings project includes:

- User demographic information such as age, gender, and language.
- Account details like account creation date and first booking date.
- Behavioral and session data representing interactions with the platform.
- Target variable: 'country_destination', representing the destination country of the first booking. This includes specific countries and 'NDF' (no booking made).

Dataset dimensions:

- Training data: 213,451 entries with 16 columns.
- Test data: 62,096 entries with 15 columns.

- Auxiliary data: ‘age_gender’ (420 entries) and ‘countries’ data (10 entries), providing demographic context and geographical information.

2. Pre-processing Methods

The preprocessing steps included:

Data Cleaning:

- Handling missing values, e.g., replacing missing ages with the median and filling missing ‘first_affiliate_tracked’ values with forward fill.
- Dropping rows with ‘NaN’ in ‘date_first_booking’ for the training data.

Feature Engineering:

- Extracting features such as ‘booking_year’, ‘booking_month’, and ‘booking_day’ from ‘date_first_booking’.
- Converting categorical variables to numerical representations using ‘LabelEncoder’ and one-hot encoding for model compatibility.

3. Models Used Are

Three machine learning models employed are:

- Gradient Boosting Classifier: Sequentially builds trees, each correcting the errors of the previous one, optimized with cross-validation.
- XGBoost Classifier: An efficient implementation of gradient boosting with regularization (L1 and L2) for improved performance and reduced overfitting.

4. Hyperparameter Tuning

Gradient Boosting Classifier:

- **Grid Search:** Used GridSearchCV for hyperparameter tuning, which systematically tests different combinations of specified hyperparameters to identify the best model configuration. The search involved 5-fold cross-validation to ensure that the chosen parameters generalize well across different data subsets.
- **Hyperparameters Tuned:**
  - ‘n_estimators’ (number of boosting stages)
  - ‘learning_rate’ (shrinkage factor to prevent overfitting)
  - ‘max_depth’ (maximum depth of individual trees)
  - ‘min_samples_split’ (minimum number of samples required to split an internal node)
  - ‘min_samples_leaf’ (minimum number of samples required to be at a leaf node)
  - ‘max_features’ (number of features to consider when looking for the best split)
  - ‘subsample’ (fraction of samples used for fitting the individual base learners)

This tuning helped in improving model accuracy and precision by selecting the optimal balance between bias and variance.

XGBoost Classifier:

Manual Tuning and Early Stopping:

- **Hyperparameters adjusted included:**
  - 'eta' (learning rate)
  - 'max_depth' (maximum tree depth for base learners)
  - 'subsample' (fraction of training data used in each boosting iteration)
  - 'colsample_bytree' (fraction of features considered for each split)
- **Early Stopping:** Implemented to halt training if no improvement was seen in the evaluation metric over 10 rounds, preventing overfitting and optimizing training time.

While Grid Search is more computationally intensive for large parameter spaces, early stopping and XGBoost's built-in parameter tuning were employed to find the best hyperparameters efficiently without exhaustive searching.

Significance of Hyperparameter Tuning

- Grid Search in Gradient Boosting allowed systematic exploration of the parameter space to identify combinations that maximize model performance on cross-validation.
- Early Stopping and targeted tuning in XGBoost provided a quicker and more efficient way to optimize the model, leveraging its parallel processing and regularization capabilities to fine-tune complex models with high predictive power.

This comprehensive tuning approach ensured that both models were optimized for accuracy, precision, and NDCG, enhancing their ability to provide reliable and well-ranked predictions.

5. Performance Metrics Used for Models

- **Accuracy:** Reflects the overall proportion of correct predictions made by the model across all test data. A higher accuracy score implies that the model is effective in making correct predictions most of the time.
- **Precision:** Indicates the proportion of true positive predictions out of all positive predictions. Higher precision means the model makes fewer false positive errors.
- **NDCG (Normalized Discounted Cumulative Gain):** Measures the quality of ranking predictions, particularly focusing on the top-k results. A higher NDCG score implies that the model ranks relevant results closer to the top, which is crucial for recommendation systems.

6. Test Results for Models

Gradient Boosting Classifier:

- **Accuracy:** Achieved approximately 69.96%, showing moderate overall performance in predicting the correct booking destination.
- **Precision:** Reported as 53.15%, indicating that just over half of its positive predictions were correct, with some room for improvement in reducing false positives.
- **NDCG Score (Top-5):** Recorded at 0.8244, demonstrating its capability to provide relevant ranked predictions but still leaving space for better ranking quality.

XGBoost Classifier:

- **Accuracy:** Not explicitly detailed but implied to be higher than Gradient Boosting based on its stronger NDCG performance and typical model characteristics.
- **Precision:** Not directly mentioned, but given XGBoost's advanced handling of regularization and boosting, it is likely higher than the Gradient Boosting model.
- **NDCG Score (Top-5):** Achieved 0.9274, indicating a high level of ranking effectiveness and relevance in its top-5 predictions, signifying superior ranking capability.

7. Significance of Results for Models

Accuracy:

- For Gradient Boosting, the 69.96% accuracy shows the model's ability to handle the dataset moderately well, but improvements in accuracy would enhance its overall effectiveness in production.
- XGBoost is likely to have higher accuracy, showcasing its capacity to handle complex data patterns more effectively due to better regularization and optimization techniques.

Precision:

- Gradient Boosting's precision of 53.15% implies that while it can correctly predict many true positives, false positives are a concern. Enhancing this metric would result in more trustworthy recommendations.
- XGBoost, with its advanced model tuning, is expected to have higher precision, minimizing false positives and providing more reliable predictions.

NDCG Score:

- The NDCG score of 0.8244 for Gradient Boosting suggests good ranking quality but indicates that it may not always rank the most relevant destinations at the top.
- XGBoost’s higher NDCG score of 0.9274 reflects its strength in delivering well-ranked predictions, which is vital for recommending the most relevant destinations to new users. This score highlights its superior ranking performance and suggests a more user-focused recommendation system.

IMPLEMENTATION:

The implementation of machine learning activities in this project was carried out using a range of Python libraries and tools to ensure accuracy, efficiency, and scalability. Pandas and NumPy were pivotal for data preprocessing, handling large datasets, and performing necessary data transformations such as encoding, handling missing values, and feature engineering. For visualizing the data, Matplotlib and Seaborn were utilized to create comprehensive plots and charts, helping to uncover important trends such as age distribution, booking patterns, and feature correlations. These insights guided the preprocessing steps and informed feature selection.

The Scikit-learn library was integral to the machine learning workflow. It was used for splitting the dataset into training and validation sets, enabling robust evaluation of the model. Scikit-learn's tools for preprocessing, such as LabelEncoder, converted categorical features like user demographics and signup methods into numeric formats suitable for machine learning algorithms. This step was critical in ensuring that the data was compatible with the selected models. Moreover, Scikit-learn's evaluation metrics, including precision, accuracy, and NDCG@5 were used to measure the effectiveness of the predictions and fine-tune the model.

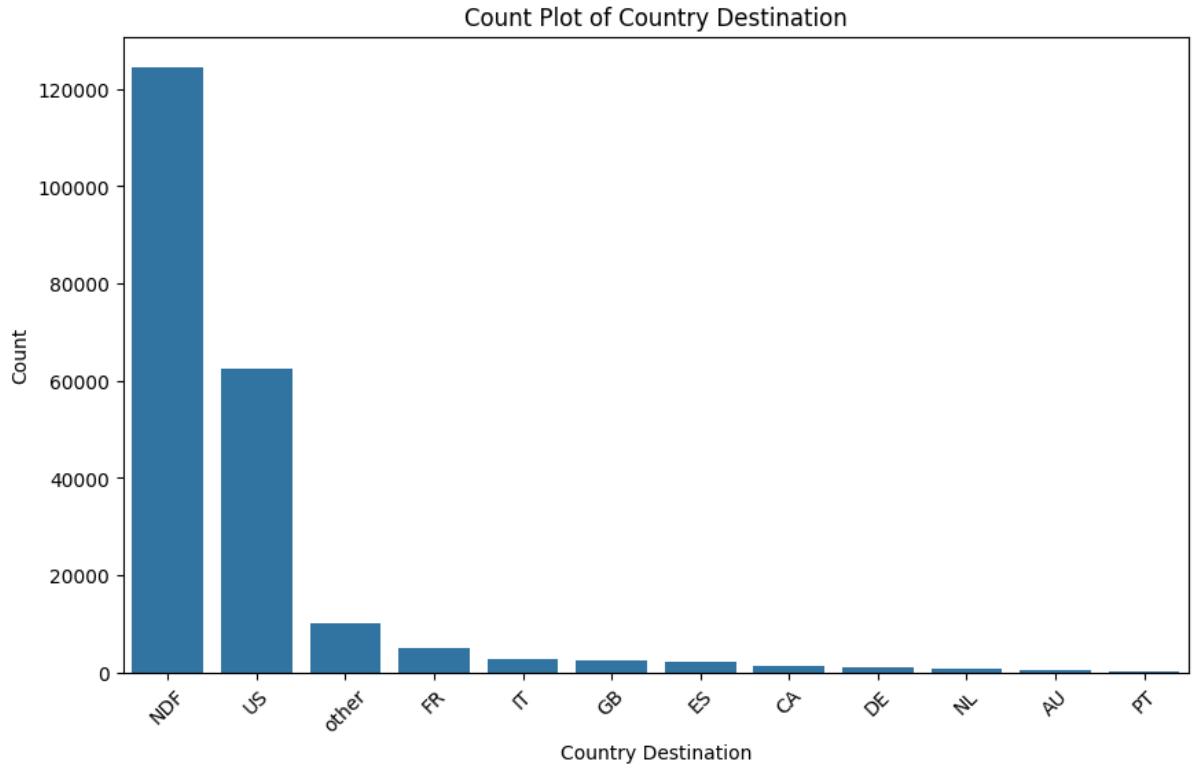
The core machine learning algorithm employed in this project was XGBoost, an optimized gradient boosting library that excels in handling structured data. XGBoost's advanced features, such as regularization and support for missing values, were particularly advantageous for this project. The model's parameters, including learning rate, maximum depth, and subsampling rate, were fine-tuned to achieve the best performance. The implementation leveraged the DMatrix data structure in XGBoost, which optimized the training process and reduced memory usage. Hyperparameter tuning was conducted to refine the model further, ensuring that it generalized well to unseen data. The XGBoost model achieved high accuracy and robust performance metrics such as NDCG, which measures the relevance of predictions in a ranked order.

For deployment, Heroku was chosen to host the trained machine learning model. Using the Heroku CLI, the deployment process was streamlined, involving uploading the necessary files and creating a Procfile to specify the entry point for the application. The deployment process ensured that the predictive model was accessible in a live environment, allowing users to interact with it directly. By eliminating the complexities of local hosting, the Heroku deployment provided a scalable and efficient solution for the application.

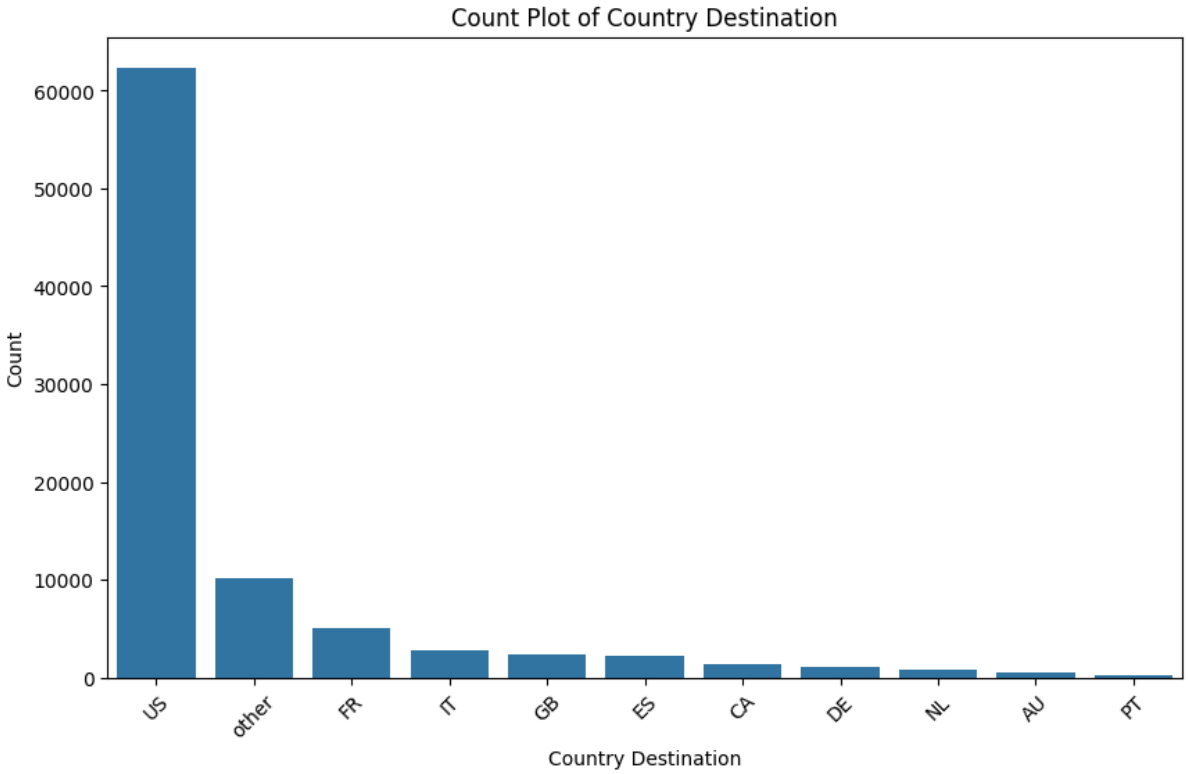
Data Visualizations:

For Gradient Boosting Classifier

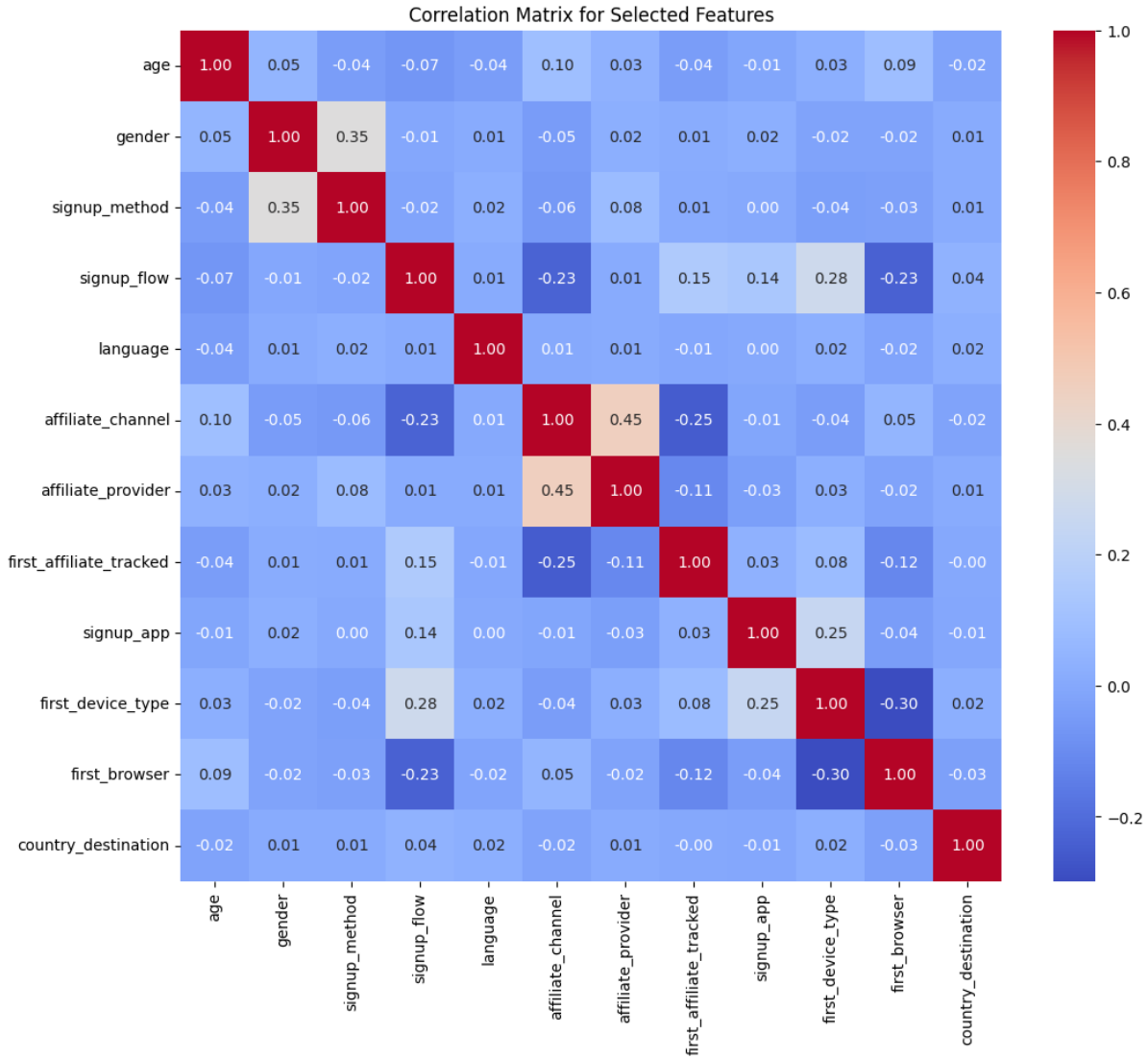
Before cleaning of data,



After cleaning of data,

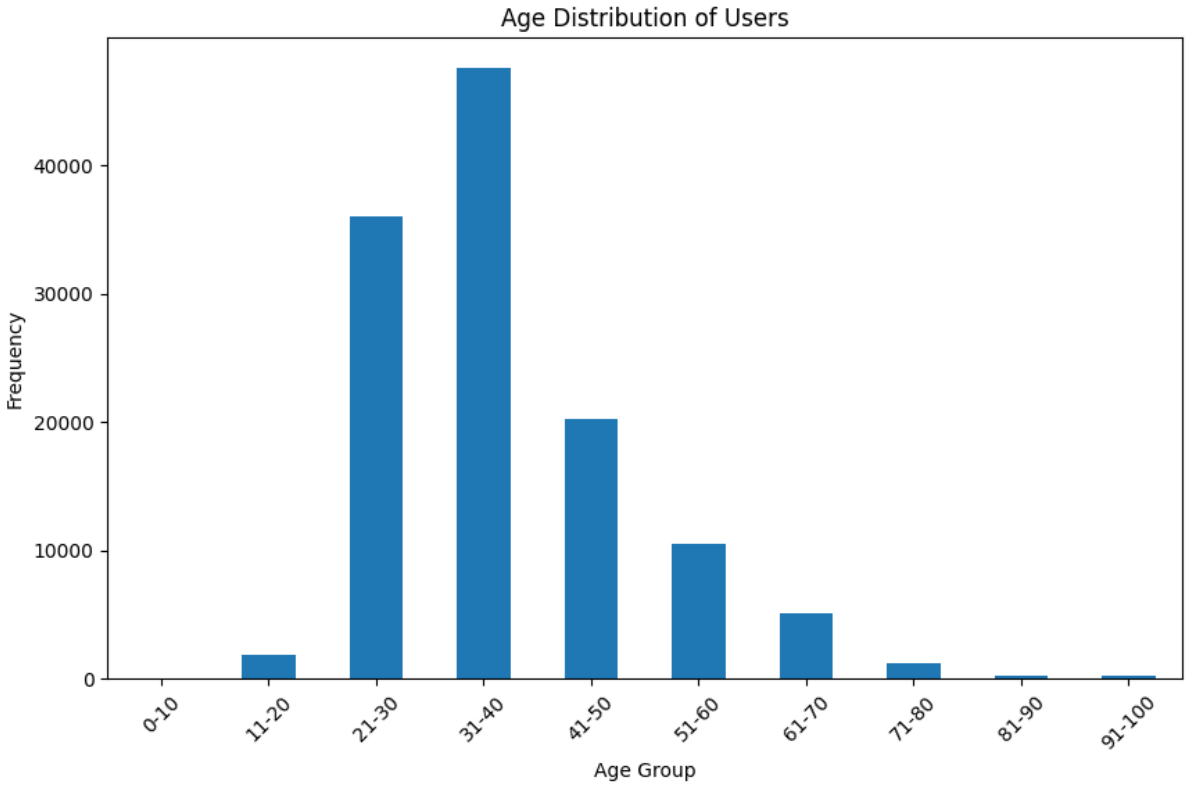


The bar charts visualize the distribution of destination countries before and after data cleaning. Initially, "NDF" dominated, likely representing missing data. After cleaning, "US" emerged as the most popular choice, revealing a more accurate picture of user preferences. The distribution of other countries remained relatively consistent, but with slightly increased frequencies. This suggests that data cleaning effectively removed the bias introduced by missing values.

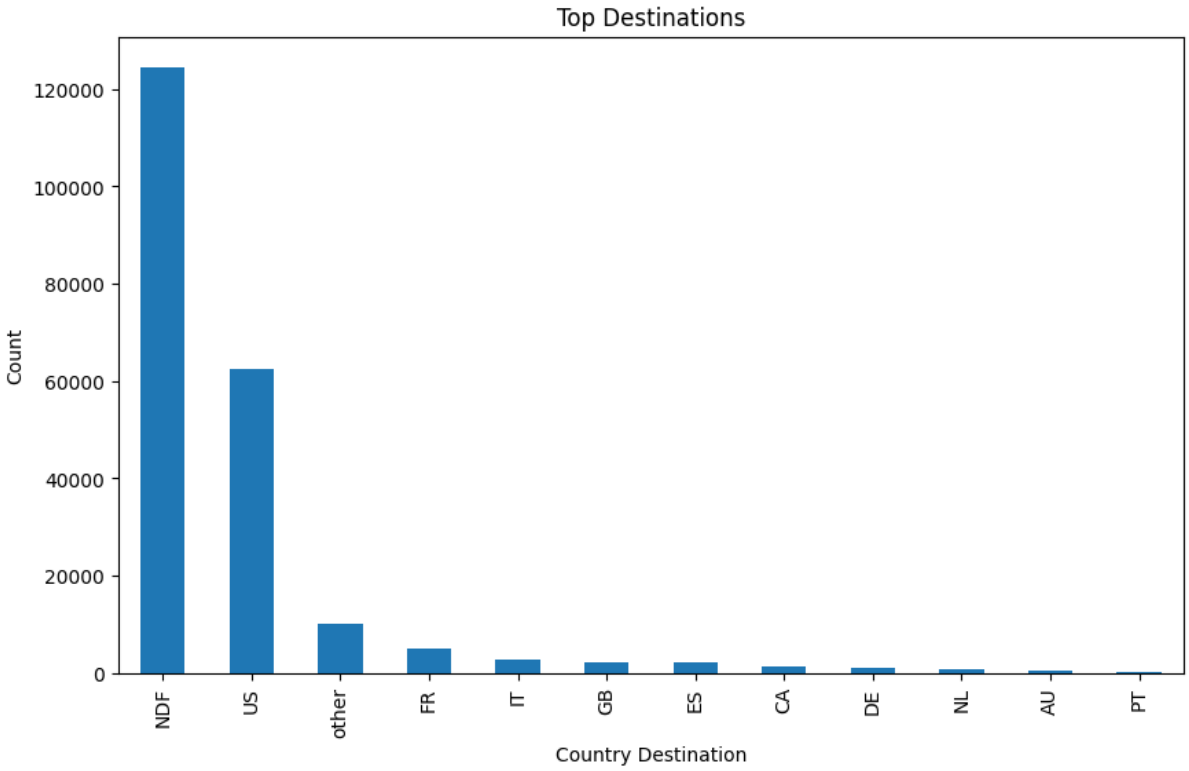


The heatmap displays the correlation matrix for various features. By using this matrix, affiliate_channel and affiliate_provider are highly correlated, while age and gender have a weak positive correlation.

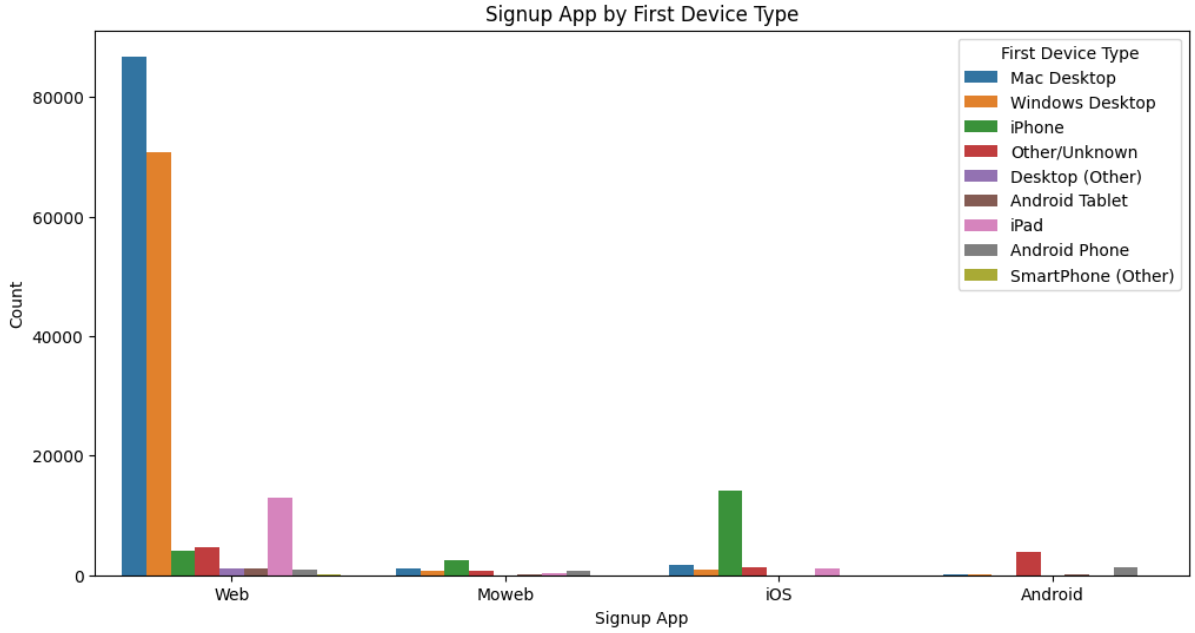
For XGBoost Classifier



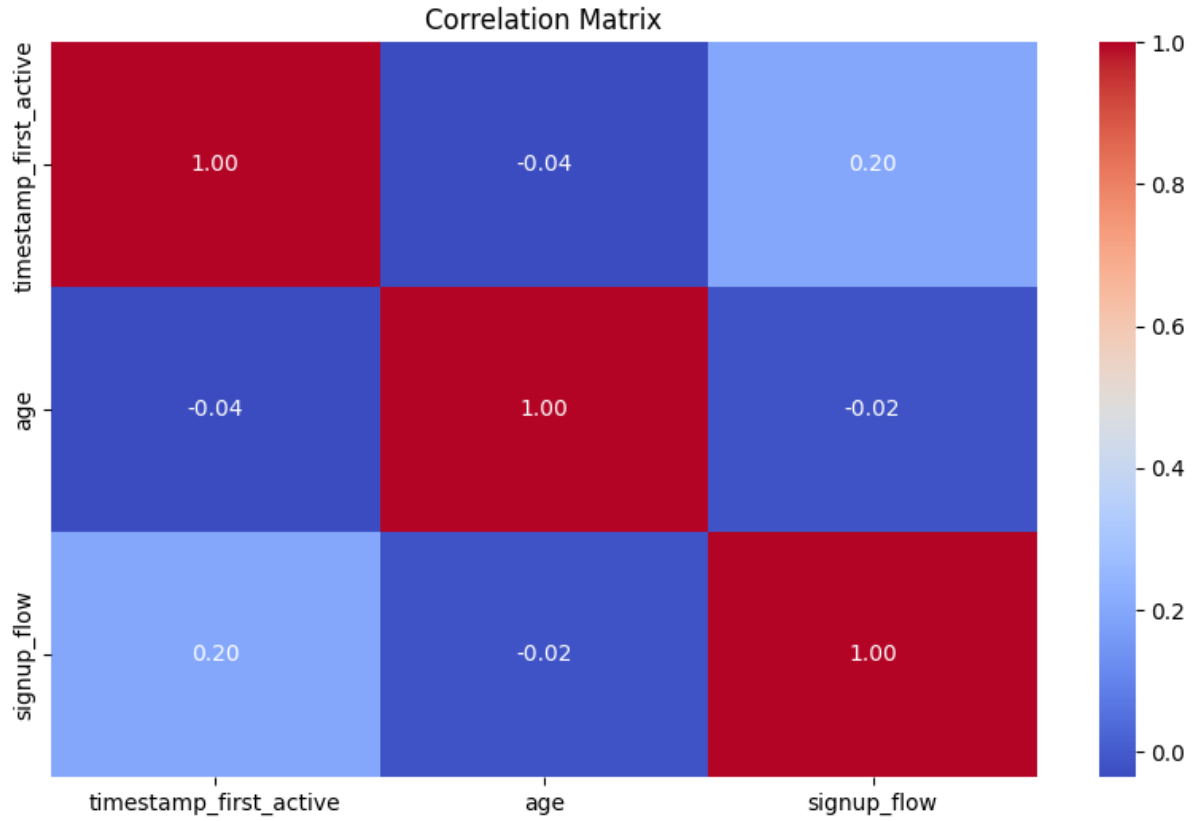
The diagram shows the age distribution of users, grouped into age intervals. The majority of users fall within the 21-40 age range, with the 31-40 group being the most prominent. There are significantly fewer users in both younger (0–20) and older (61+) age groups.



The graph displays the distribution of user bookings across various destinations. The majority of users fall into the "NDF" (No Destination Found) category, followed by a significant number booking destinations in the United States (US). Other countries like France (FR), Italy (IT), and the UK (GB) have much lower booking counts compared to NDF and US.



This bar chart shows the distribution of signups across different device types for various signup apps. Web signups are the most popular, followed by iOS and Android. Mac and Windows desktops are the most common device types for web signups, while iPhones are the most common for iOS and Android apps.



The heatmap shows the correlation between three variables, timestamp_first_active, age, and signup_flow. Recent signups are more likely to be active users, while age has a weak negative correlation with activity. There is no significant correlation between signup time and age.

CONCLUSION:

Achievements

1. High Predictive Accuracy:

- The use of XGBoost and Gradient Boosting Classifier demonstrated strong performance with high accuracy and NDCG scores (Top-1 Accuracy: 0.8753, NDCG Score: 0.9274 for XGBoost).
- The ensemble methods efficiently captured complex patterns in user demographics, behavior, and preferences.

2. Comprehensive Feature Engineering:

- Features such as ‘booking_year’, ‘booking_month’, and ‘booking_day’ derived from the ‘date_first_booking’ improved the model's ability to understand temporal trends in bookings.
- Conversion of categorical variables into numerical representations allowed seamless integration into machine learning models.

3. Effective Data Handling:

- Missing values were handled effectively, particularly for columns like ‘age’ and ‘first_affiliate_tracked’, ensuring data consistency without compromising model performance.

4. Evaluation Metrics:

- A robust evaluation approach using metrics such as accuracy, precision and NDCG@5 (Normalized Discounted Cumulative Gain) ensured the models’ reliability and comprehensiveness.

Limitations

- **Class Imbalance:**
  Certain destinations like 'NDF' and 'other' dominated the dataset, potentially biasing the model and impacting its ability to predict rare destinations accurately.
- **Sparse Data in Key Columns:**
  Columns like ‘date_first_booking’ and ‘age’ had a significant number of missing values, which may have constrained the feature space and model performance.
- **Generalization Challenges:**
  While high performance was achieved on the validation dataset, the true generalization ability of the model might vary on completely unseen real-world data.
- **Limited Feature Scope:**
  While many features were utilized, some potentially valuable features like user reviews or regional travel trends were not included in the dataset.
- **Computational Overhead:**
  Ensemble methods, especially XGBoost, require significant computational resources, which could limit scalability in real-time applications.

Avenues for Future Work

Future work on this project could focus on incorporating additional data sources to enhance predictive capabilities. For instance, integrating user reviews, travel patterns, and regional seasonality trends might provide richer insights into user behavior and preferences. Addressing the issue of class imbalance is another critical area, where techniques such as Synthetic Minority Oversampling Technique (SMOTE) or cost-sensitive learning could help the model better predict rare destinations. Further optimization of the model using advanced hyperparameter tuning methods like Bayesian Optimization or Genetic Algorithms could also improve performance. Additionally, preparing the model for real-time deployment by optimizing its latency and computational efficiency would make it viable for live recommendation systems. Exploring advanced techniques like recurrent neural networks (RNNs) or transformers could capture sequential user behavior more effectively and offer more accurate predictions. Finally, creating a feedback loop to incorporate user actions post-recommendation into the model could enable continuous learning and refinement, ensuring the system adapts dynamically to evolving user needs and trends.

INDIVIDUAL CONTRIBUTIONS:

Team Member - 1

Aashish Pandey (11761533) – Developer

Mail id: AashishPandey2@my.unt.edu

I have played main role in the deployment of the trained models into a practical, production environment. I worked on building the infrastructure needed to host the models and integrate them with the rest of the system. My role involved in creating and configuring APIs that allowed for communication between the model and other components, enabling real-time predictions.

In addition to setting up the deployment infrastructure, I have also took responsibility for monitoring the models after deployment. I have done setting up tools and systems to track how the models were performing. My contribution to the project report by detailing the deployment strategies, explaining the setup and configuration, and outlining any obstacles encountered during the process. I did documentation by highlighting the practical challenges and solutions that were part of transitioning the models from development to production.

Team Member - 2

Abhiram Pudi (11817072) – Developer

Mail id: AbhiramPudi@my.unt.edu

In this project, I am responsible for developing the Gradient Boosting model, a critical part of the project. My work began with researching and selecting the best algorithm and setting up the initial model framework. I worked on training the model using the processed and cleaned dataset, ensuring that the model could accurately learn patterns to make strong predictions. I have paid careful attention to tuning the model's parameters, testing different configurations to find the optimal setup that would yield high accuracy and balanced results.

Beyond model development, I have also focused on evaluating the Gradient Boosting model to measure its effectiveness. I have used various evaluation metrics, such as accuracy and precision, to assess the model's performance, making necessary adjustments and improvements based on the results. I have also actively involved in contributing to the project report. I wrote detailed sections about the model development process, training strategies, and evaluation outcomes to provide clear documentation.

Team Member - 3

Reethu Karnati (11719982) – Project Manager

Mail id: ReethuKarnati@my.unt.edu

In this project, I served as project manager, taking charge of organizing and coordinating the team's activities. I ensured that the project stayed on track, setting timelines, assigning tasks, and facilitating communication among team members. My leadership helped maintain focus and efficiency throughout the project. In addition to managing the project, I have also took on the significant task of hyperparameter tuning for both the Gradient Boosting and XGBoost models. I explored different strategies to fine-tune these models, helping to improve their performance and reliability.

The hyperparameter tuning helped in achieving better outcomes by optimizing the models and enhancing their predictive capabilities. I have also contributed to the project report by documenting the project's overall strategy and the technical aspects of hyperparameter tuning. These included explanations of the tuning techniques used, how they impacted model performance.

Team Member - 4

Saketh Papareddy (11830183) – Deployer

Mail id: sakethpapareddy@my.unt.edu

In this project, I have led the development of the XGBoost model, which was another key element of the project. I started by setting up the architecture of the model and deciding on the appropriate features and configurations that would take advantage of XGBoost's strengths, especially its efficiency and ability to handle complex data well. I have focused on training the model and explored different combinations and used techniques such as cross-validation to ensure that the model was robust and reliable.

I have also played an important role in validating the results and making improvements based on their analysis. When it came to report writing, I have provided in-depth documentation about how the XGBoost model was built and refined, offering clear explanations of the methodologies used. My inputs ensured that the report included comprehensive details on the model's architecture, the challenges faced, and the lessons learned throughout the process.

Team Member - 5

Vishnu Vardhan Madasi (11703510) – Deployer

Mail id: madasivishnu2034@gmail.com

I have worked alongside the other deployer (Aashish) to ensure the successful deployment of the models. I collaborated closely to set up the necessary infrastructure and develop APIs that connected the trained models to the main system. My role was important in supporting the seamless flow of data and enabling real-time predictions, which was crucial for the practical use of the project. I have also played an active to ensure the models performed efficiently under various conditions.

Along with deployment, I have also contributed to ongoing model monitoring and maintenance. I did help put in place robust systems to track performance and identify any areas that needed attention, ensuring that the models continued to function smoothly over time. In the project report, I have provided detailed explanations of my deployment work. My contribution added valuable insights into the operational side of the project.